International Journal of Applied Science and Engineering
2019. 16, 2: 119-131

Special Issue of The 13th International Conference on Advanced Information Technologies (AIT 2019)

Interpolation Data Hiding Using Dual Folding Methodology

Tzuchuen Lu*, Shiru Huang and Thanhnhan Vo

Deparment of Information Managerment, Chaoyang University of Technology, Taichung Taiwan, R.O.C

Abstract: Interpolation is an important technique in digital image processing. This technique allows of reconstructing the intermediate pixels that are distorted in the process of stretching image appropriately. Therefore, this is a technical branch in the image hiding technique and it is considered to apply in steganography. Neighbor Mean Interpolation (NMI) technique was proposed by Jung and Yoo and later was improved on the Interpolation of Neighbouring Pixels (INP) technique by Lee and Huang to be more effective in securing the confidential information to third parties. Lee and Huang have improved the payload of 56% up to 108% from Jung and Yoo's method. However, both two methods have relatively low image quality. The strategy of Lu et al. has high effect in improving PSNR and payload because the basis of this technique is to minimize the value of the secret data before hiding them into the cover image. The proposed scheme applies Lu et al.'s central folding strategy twice to effectively enhance the PSNR and payload parameters compared with (NMI) and the Interpolation of Neighbouring Pixels technique (INP).

Keywords: Reversible data hiding; interpolation; central folding method; secret symbol.

1. Introduction

Data hiding technique is an important research issue in encryption and information security. The secret message as text, image, audio, video, and so on need to be guaranteed safety in the storage and information communication via Internet. Therefore, researchers embed the secret message into the cover image to form the stego image for secret sharing. Stego image is sent to the receiver. These actions ensure that the secret message is strongly protected from the third party. The receiver, who knows the decryption method can extract the message from the stego image. The receiver will conduct the reverse process called extracting processing to obtain the secret message from the stego image. In steganography, there are two general techniques used to embed data into the cover images called reversible data hiding (RDH) [1, 2, 3] and irreversible data hiding [4, 5]. In the RDH method, the receiver can extract the message from the stego image that retains the original image without damaged. This is necessary when it is applied in the military, and the medicine. In the irreversible data hiding method, the original image cannot be reconstructed.

A further problem is when we hide the data into the cover image. The resolution of the original image is reduced. The naked eyes can distinguish the changes of the stego image compared with the original image. Therefore, image interpolation is a way to solve this problem. Interpolation technique is used to compute the reasonable value of the intermediate pixels when we stretch the cover image and embed the secret message in it without breaking the image quality [6, 7, 8, 9, 10, 11].

Corresponding author; e-mail: tclu@cyut.edu.tw
Received 1 May 2019
doi: 10.6703/IJASE.201909_16(2). 119
Revised 22 August 2019

Accepted 29 August 2019

Jung and Yoo [12] introduced a data hiding method using an interpolation technique, called Neighbor Mean Interpolation (NMI). This method was improved later by Lee and Huang, Interpolation by Neighbor Pixels (INP) [13]. However, the PSNR value of the two methods is quite low. In this paper, we proposed an enhanced hiding scheme by using center folding strategy. The center folding strategy is proposed by Lu et al. in 2015 [8]. This method is effective in improving the PSNR value and the embedding capacity. Therefore, this scheme adopted the strategy to reduce the image distortion.

The proposed hiding method using the interpolation technique combined with the center folding strategy. The proposed method will present in the following sections sequentially: Section 2 describes the methods relate to this study, included: Neighbor Mean Interpolation (NMI) method (Section 2.1), Interpolation by Neighboring Pixels (INP) (Section 2.2), Center Folding Strategy (Section 2.3). Section 3 describes the proposed method. Finally, Section 4 concludes the propose method.

2. Related Works

2.1 Neighbor Mean Interpolation (NMI)

Jung and Yoo (2009) announced a method using the interpolation technique in digital image processing. Figure 1 presents the overall diagram of Jung and Yoo's scheme. The input image with size $w \times h$ was scaled down to be size $(w/2) \times (h/2)$ image which is called the original image. Then they used the interpolation technique to interpolate the original image and become the cover image with size $w \times h$, respectively. This cover image is used for the embedding process. The secret data is embedded into the cover image to produce the stego image. The receiver can extract the secret data and reconstruct the original image from the stego image.

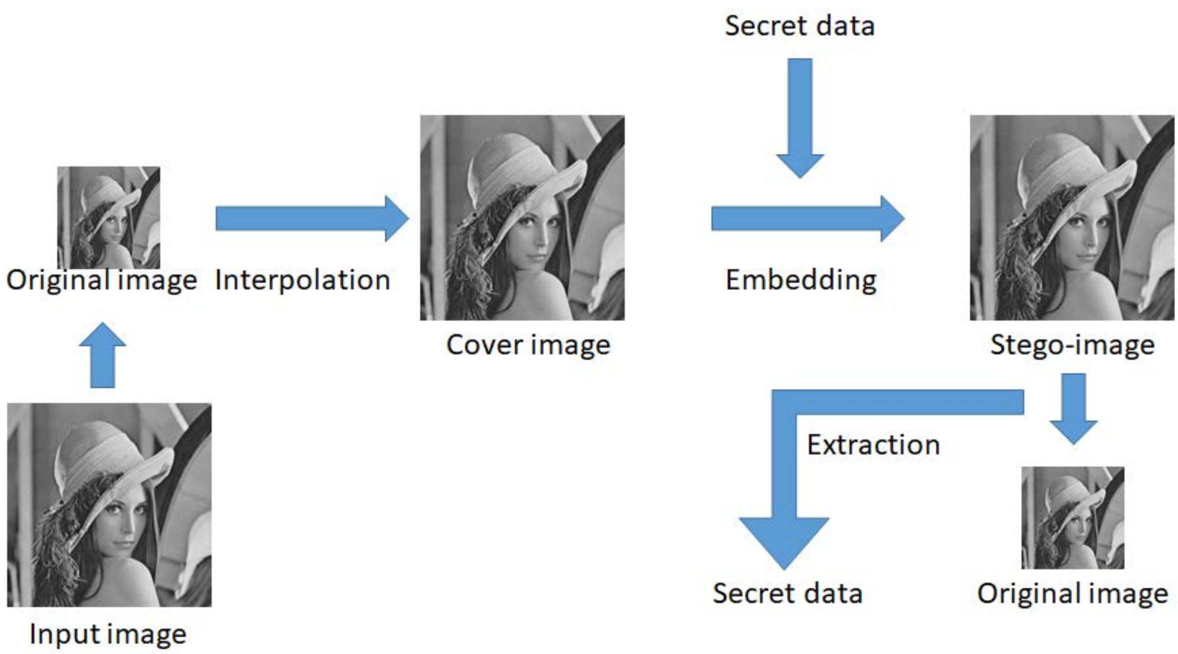

Figure 1. The flowchart of Jung and Yoo's method.

Jung and Yoo (2009) created the NMI method. Firstly, they scaled the input image size down four times to make the original image. For example, an input image with 512×512 pixels will be scaled down into an image with 256×256 pixels to make the original image (*I*). The NMI technique was used to interpolate and produce the cover image (*C*) with size 512×512. This interpolation method is implemented with the rules by calculating the mean value of the adjacent pixels that obtain the value of the intermediate pixels as showing in Eq.(1). Figure 2 shows the interpolation of a block of the original image from size 2×2 to size 3×3. Four corner pixels in Figure 2(b) are respectively equal the initial pixels in Figure 2(a). The rest pixels of the block pixels in Figure 2(b) are obtained using Eq.(1), where $0 \le j \le i$ and $m, n = 0,1, \dots ,127$.

$$C(i,j) = \begin{cases} I(i,j), & if\ i = 2m, j = 2n, \\ (I(i,j-1) + I(i,j+1))/2, & if\ i = 2m, j = 2n+1, \\ (I(i-1,j) + I(i+1,j))/2, & if\ i = 2m+1, j = 2n, \\ (I(i-1,j-1) + C(i-1,j) + C(i,j-1))/3, & otherwise \end{cases} \quad (1)$$

Where, the pixel $C(i,j) = I(i,j)$ for $i = 2m, j = 2n$ is referred as a axis pixel which will never be changed in the embedding processing. Assume that $I(x,y)$ represents the value of a pixel located at (x,y) in the original image *I*. After the NMI method is implemented, a cover pixel $C(i,j)$ at location (i,j) of the cover image *C* and the symbol $C(i,j)$ perform the value of a stego-pixel at location (i,j). Then the secret message is embedded into the cover image. The procedure of NMI is described as follows.

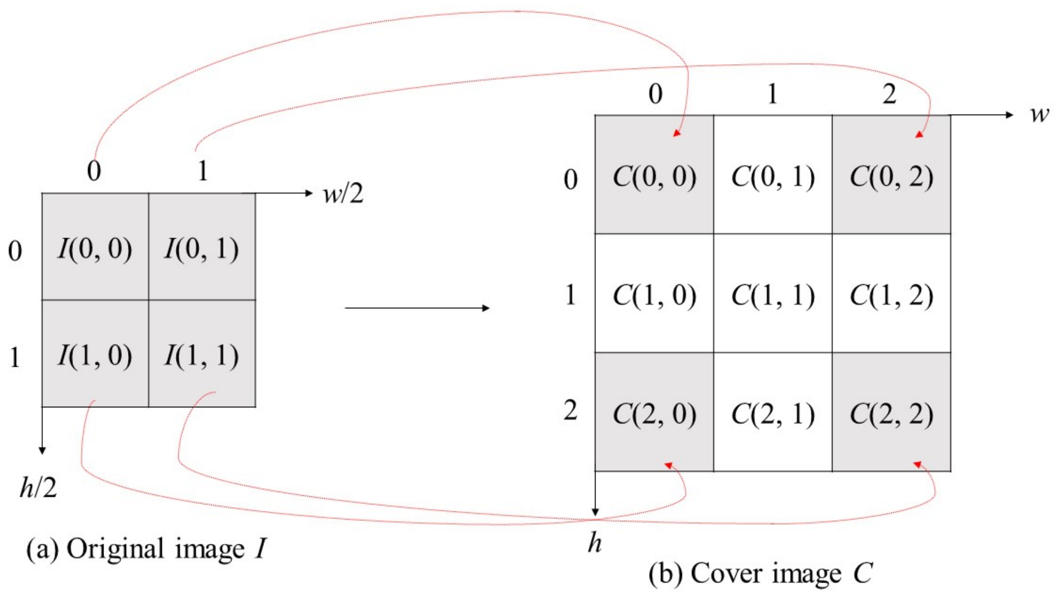

Figure 2. The cover image generated from the original image by calculating NMI.

2.1.1 Procedure of Neighbor Mean Interpolation (NMI)

Input:

An original image *I* with size $(w/2)\times(h/2)$, where $I(x,y)$ represent the value of a pixel location at (x,y) in the original image I, $x = 0, 1, \ldots, (w/2) - 1$; $y = 0, 1, \ldots, (h/2) - 1$.

Output:

A cover image *C* with $w \times h$ size is produced.

Step 1. Cover pixels are calculated as follows.
Initially, set index variables $i = j = 0$ and $x = y = 0$.
Assign each cover pixel of a 2×2 block as follows Eq.(1).

Step 2. If $j + 2 < w$, then
$j = j + 2, y = y + 1$, and go to Step 1
else proceed with Step 3.

Step 3. If $j + 2 = w$ and $i < h$, then
$i = i + 2, j = 0, x = x + 1, y = 0$, go to Step 1.
else proceed with Step 4.

Step 4. A cover image *C* with size $w\times h$ is produced.

2.2 Interpolation by Neighboring Pixels (INP)

This method was improved from Jung and Yoo's strategy (NMI) by Lee and Huang [13]. The intermediate pixels is calculated by referring to the neighbor pixels. The INP method effectively exploits the characteristics of the neighbor pixels to interpolate. Thus, giving the cover image with same average PSNR values, but archiving a good embedding rate. The payload of the INP method improved by up to 56% to 108% when both NMI and INP methods experimented on the same input images. The calculation of the INP method is performed according to Eq. (2) as follows, where $0 \le j \le i$ and $m, n = 0, 1, \ldots, 127$.

$$C(i,j) = \begin{cases} I(i,j), & if\ i = 2m, j = 2n, \\ (I(i,j-1) + (I(i,j-1) + I(i,j+1))/2)/2, & if\ i = 2m, j = 2n+1, \\ (I(i-1,j) + (I(i-1,j) + I(i+1,j))/2)/2, & if\ i = 2m+1, j = 2n, \\ (C(i-1,j) + C(i,j-1))/2, & otherwise \end{cases} \quad (2)$$

Where the pixel $C(i,j) = I(i,j)$ for $i = 2m, j = 2n$ is referred as an axis pixel which will never be changed in the embedding processing. Assume that $I(x,y)$ represents the value of a pixel located at (x,y) in the original image *I*. After the INP method is implemented, a cover pixel $C(i,j)$ at location (i,j) of the cover image *C* and the symbol $C(i,j)$ perform the value of a stego-pixel at location (i,j). Then the secret message is embedded in the cover image. The procedure of INP method is described as follows.

2.2.1 Procedure of Interpolation by Neighboring Pixels (INP)

Input:

An original image I with size $(w/2)\times(h/2)$, where $I(x,y)$ represent the value of a pixel location at (x,y) in the original image I, $x = 0, 1,..., (w/2) - 1; y = 0, 1, ..., (h/2) - 1$.

Output:

A cover image C with size $w\times h$ is produced.

Step 1. Cover pixels are calculated as follows.
Initially set index variables $i = j = 0$, $x = y = 0$.
Assign each cover pixel of a 2×2 block as follows Eq. (2).

Step 2. If $j + 2 < w$, then
$j = j + 2$, $y = y + 1$, and go to Step 1
else proceed with Step 3.

Step 3. If $j + 2 = w$ and $i < h$, then
$i = i + 2$, $j = 0$, $x = x + 1$, $y = 0$, go to Step 1.
else proceed with Step 4.

Step 4. A cover image C with size $w\times h$ is produced.

2.3 Center Folding Strategy

We realize that both of NMI and INP only solve the problem of interpolation to create a cover image. Although the INP method is more advanced than the NMI method, for the higher payload, but the PSNR value has not improved yet. Therefore, we combined the INP method with a center folding method [8] to improve the PSNR value and the capacity.

Lu et al. found that when the secret data is embedded much, it will affect a lot of image quality, and also limit the embedding ability when it reaches a certain limit. If the value of the secret data is too large, it can cause huge distortion between stego-pixels and original-pixels. Lu et al. proposed the center folding method to reduce the value of the secret symbol. After the secret symbol values are reduced, they are embedded in stego image to obtain better embedding quality. Thereby improving image quality and reducing the executed time in the embedding process.

3. Proposed Method

In this paper, we used two phases in data hiding process to increase the value of PSNR and capacity factors. The first phase is to create a cover image (C) from the original image (I) by applying the INP method to encode and create the cover image to repairing for the embedding process. The second phase, the secret message has reduced the value of the secret symbols and they are embedded in the embedding process by using the center folding technique of Lu et al. The results are shown in the experiment results section and the conclusion section. The results of our method have higher PSNR value and better capacity than the NMI method and INP method. The scheme of the proposed embedding method is shown as follows.

3.1 Embedding procedure

The scheme of the proposed embedding method is shown as follows.

Step 1. The INP of hiding data method was used to produce the cover image of 3×3 blocks from 2×2 blocks and calculate the value of the pixels in 3×3 blocks as in Figure 3. This process is as follows.

Step 2. Using the center folding method to encode the secret message, this purpose is to reduce the value of the secret message before embedding in the cover image, where k is the number bits that were taken as a set to convert into the secret symbol d $(d \in R)$. We set, the values of d are shown in Figure 4.

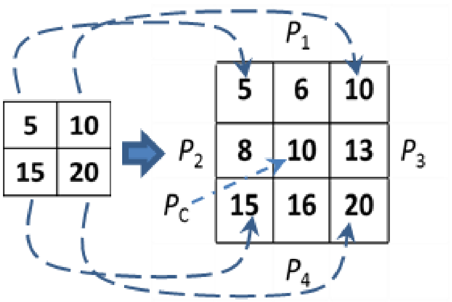

Figure 3. An example describes for the proposed scheme.

Where P_1, P_2, P_3, P_4, P_c are calculated by

$$P_1 = \left\lfloor \frac{5\times2+10}{3} \right\rfloor = 6$$

$$P_2 = \left\lfloor \frac{5\times2+15}{3} \right\rfloor = 8$$

$$P_3 = \left\lfloor \frac{10\times2+20}{3} \right\rfloor = 13$$

$$P_4 = \left\lfloor \frac{15\times2+20}{3} \right\rfloor = 16$$

$$P_C = \left\lfloor \frac{5\times10+15}{3} \right\rfloor = 10$$

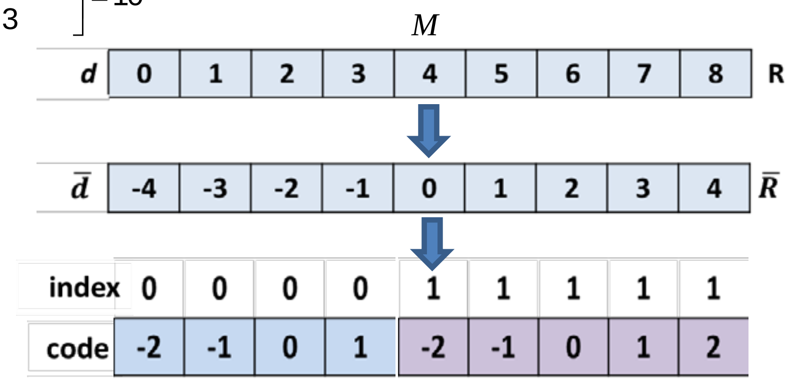

Figure 4. The center folding method.

Where M is the center position, $M = 2^{k-1} = 4$, and $\bar{R} = R - M$. And then folding one more times, we produce the values of code band in Figure 4. The necessary values after folding that we obtained to present in Table 1.

Table 1. The folded results of the secret message.

R	0	1	2	3	4	5	6	7	8
index	0	0	0	0	1	1	1	1	1
code	-2	-1	0	1	-2	-1	0	1	2

Step 3. Based on the value of the pixels that created in Step 1, including P_1, P_2, P_3, P_4, P_C. Embed the secret data after being encoded by center folding method (look up the Table 1), suppose that is the secret data $b = \{b_1, b_2, b_3, b_4\}$ and then embed b into the intermediate pixels as in Figure 5.

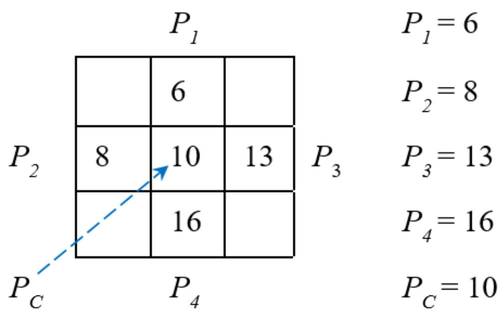

Figure 5. The positions that the secret data b embed in.

Secret data are processed in this step by using the center folding strategy so that the range of the secret symbols is $R = \{0, 1, 2, \ldots, 2^k\}$. After folding R changed to $\bar{R} = \{-2^{k-1}, -2^{k-1} + 1, \ldots, -1, 0, 1, \ldots, 2^{k-1} - 1, 2^{k-1}\}$. The formula is as follows.

$$\bar{R} = R - M \tag{3}$$

Where M is the center position, $M = 2^{k-1} = 4$, from Eq. (3) was described in Figure 4 with $k = 3$ in this example, d is the secret symbol and $\bar{d}$ is the folded secret symbol by using Eq. (3). When $d < 2^{k-1}, \bar{d}$ was shown as a negative number. When $d = 2^{k-1}$, $\bar{d}$ was shown as 0. When $d > 2^{k-1}, \bar{d}$ was shown as a positive number.

The secret data $b = \{b_1, b_2, b_3, b_4\}$ that embed in the cover image assume as $b = \{7, 5, 0, 8\}$. This data is converted from the secret data range R to the folded secret data range $\bar{R}$ by the way every 3 bits that will be retrieved, respectively, and converted from binary to decimal number. These decimal numbers are encoded as follows.

Look up Table 1 and encode $b_1 = 7$ to become $(index_1, code_1) = (1,1)$. Some example are shown bellow.

$b_2 = 5$ $(index_2, code_2) = (1,-1)$,
$b_3 = 0$ $(index_3, code_3) = (0,-2)$,
$b_4 = 8$ $(index_4, code_4) = (1,2)$.

The total bits were embedded are $3\times3+4=13$ bits, these are $\{7,5,0,8\} = \{1111010001000\}$.

Step 4. Based on $index_1$, $index_2$, $index_3$, $index_4$ that was arrange consequently according to the order of indexes and we gained $I = (1101)_2 = (13)_{10}$. Then, we calculate $\hat{I}$ from I by Eq. (4) as belows.

$$\hat{I} = I - 2^k = 13 - 8 = 5 \tag{4}$$

Step 5. Now, encode for the center pixel P_C to gain P'_C though Eq. (5)

$$P'_C = P_C + \hat{I} = 10 + 5 = 15 \tag{5}$$

Step 6. From b_1, b_2, b_3, b_4 and the value of pixels P_1, P_2, P_3, P_4 we embed P'_1, P'_2, P'_3, P'_4 accoding to Eq. (6) as follows.

$$P'_i = P_i + code_i \tag{6}$$

Where,

$P'_1 = 6 + 1 = 7,$
$P'_2 = 8 + (-1) = 7,$
$P'_3 = 13 + (-2) = 11, and$
$P'_4 = 16 + 2 = 18.$

Finally, we obtain the stego image in Figure 6 as follows.

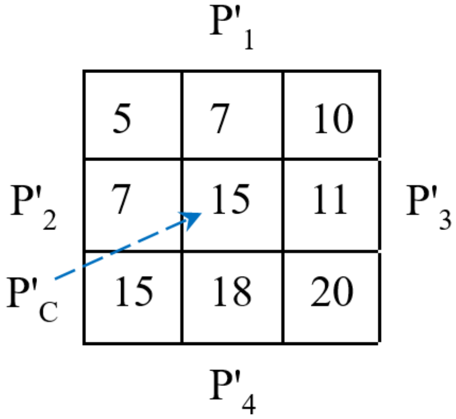

Figure 6. The stego image of the proposed method.

3.2 Extracting procedure

The recipient receives the cover image $\hat{P}$, the stego image P' and k value. The recipient combines them to generate the necessary factors to recover the original image by using two phases, firstly using the center folding method and then using the interpolation method. The extracting process was implemented as follows.

Step 1. This cover image was composed from Eq. (2)

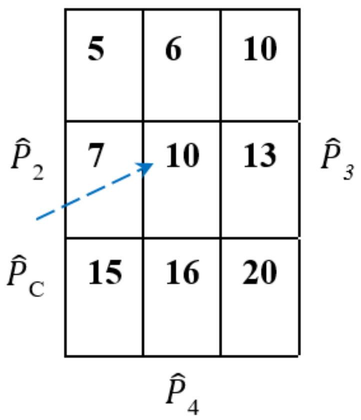

Figure 7. The cover image was sent by the sender.

Step 2. From $\hat{P}_i$ and P_i', $i = 1, 2, 3, 4$. We decoded as belows.

$$code_i' = P_i' - \hat{P}_i \tag{7}$$

Therefore,

$$code_1' = 7 - 6 = 1$$
$$code_2' = 7 - 7 = 0$$
$$code_3' = 11 - 13 = -2$$
$$code_4' = 18 - 16 = 2$$

Step 3. The indexes are generated by $\hat{P}_C$ and P_C' as belows.

$$I' = P_C' - \hat{P}_C = 15 - 10 = 5 \tag{8}$$

Then, we calculate I by Eq. (9)

$$I = \hat{I} + 2^k = 5 + 2^3 = 13 \tag{9}$$

We gained $I = (13)_{10} = (1101)_2$. The secret message were extracted and there were {1,1,0,1}.

Step 4. Combine the values of index and code in each pair, respectively. Use Table 1 to recover the secret message in decimal value.

(1,1) = 7; (1,-1) = 5; (0,-2) = 0; (1,2) = 8.

Here, the extracting process finished. We obtain $b = \{7, 5,0,8\} = \{1111010001000\}$

4. Experimental Results

Two gray scale images with size 512×512 were used to test the performance of the proposed scheme. The experiment images are shown in Figure 8.

(a) Goldhill (b) Zelda

Figure 8. Two 512×512 greyscale experimental images.

Two measurements Bits Per Pixel (bpp) and Peak Signal to Noise Ratio (PSNR) were used to judge the hiding payload and image quality of the stego-image, respectively. The equations are

$$\text{bpp} = \frac{\text{capacity}}{h \times w}, \tag{10}$$

$$\text{MSE} = \frac{\sum_{i=1}^{h}\sum_{j=1}^{w}(P_{i,j} - P'_{i,j})^2}{h \times w}, \tag{11}$$

$$\text{PSNR} = 10 \times log_{10}\frac{255^2}{\text{MSE}}\ (\text{dB}). \tag{12}$$

In the equation, $h \times w$ is the size of the original image, $P_{i,j}$ denotes the cover pixel image, $P'_{i,j}$ is the stego-pixel, MSE is the mean square error between the original image and the stego-image. Higher PSNR higher image quality is. PSNR value is higher than 35dB that means image distortion is invisible. Table 2 and Figure 9 shows the experimental results of the proposed scheme and two comparison schemes INP and NMI. In the proposed scheme, the constant k is a key factor that effect the hiding performance. In Table 2, the hiding capacities of the proposed scheme with k=2 and k=3 are 0.96 and 1.35 bpp that of NMI and INP are 1 and 1.15 bpp for the image Zelda. The image quality of the proposed scheme with k=2 and k=3 are 44.01 and 43.27 dB that of NMI and INP are 38.52 and 37.86 dB. The proposed scheme with large k can get higher hiding capacity but lower image quality. However, the hiding performance of the proposed scheme is better than that of NMI and INP.

Table 2. The comparisons of the proposed scheme and other schemes.

Images	Proposed Method				NMI		INP	
	k=2		k=3		PSNR	bpp	PSNR	bpp
	PSNR	bpp	PSNR	bpp				
Zelda	44.01	0.96	43.27	1.35	38.52	1.00	37.86	1.15
Goldhill	43.96	0.96	43.23	1.35	35.80	1.10	34.95	1.25

From Figure 9 we can see that all curves of the proposed scheme are higher than that of the other methods. That means the proposed scheme can have better image quality under the same hiding capacity.

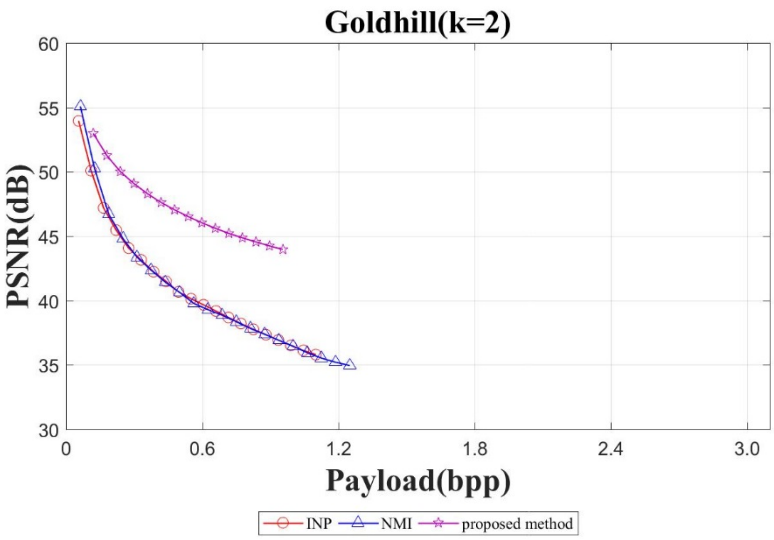

Figure 9. The comparisons of the proposed scheme and other methods with different *k*.

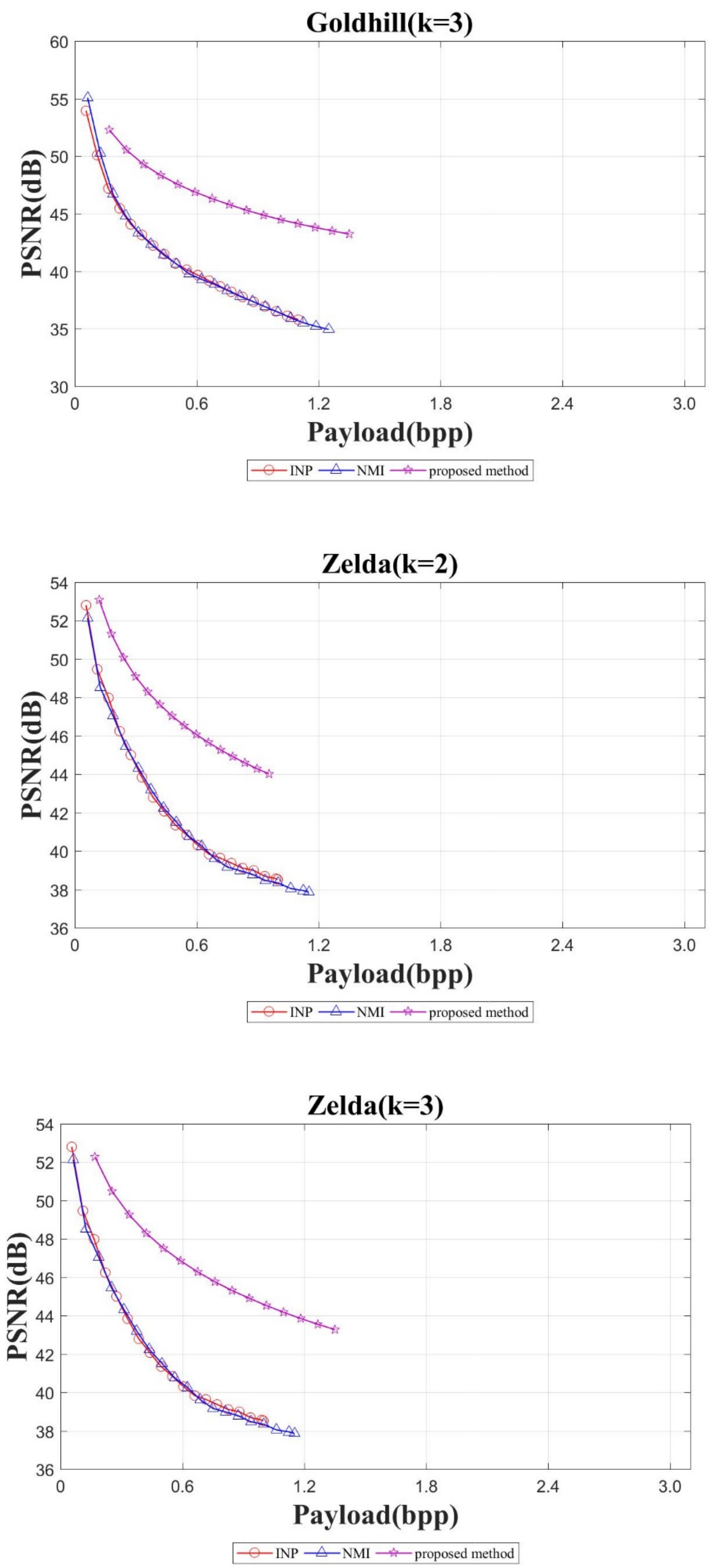

Figure 9. The comparisons of the proposed scheme and other methods with different *k*.

5. Conclusions

In general, the interpolation methods try to hide the secret data directly into the virtual pixels. However, they cost a lot of pixels to use to encode data, but in this proposed method, we use the center folding data strategy to narrow down the range of the secret data value, because of the effect of the center folding strategy of Lu et al. In this paper, we used the interpolation method and try to narrow down the secret data twice that similar mean we solved it twice by the center folding strategy, so the value of secret data become very small. The stego image that is produced by the embedding process is high quality.

References

[1] Allatar, A. M. 2004. *Reversible watermark using the difference expansion of a generalized integer transform.* IEEE Transactions on Image Processing, 13 : 1147-1156.

[2] Celik, M. U., Sharma, G., Tekalp, A. M. and Saber, E. 2002. *Reversible data hiding*. Proceedings of International Conference on Image Processing, 157-160.

[3] Tian, J. 2003. *Reversible data embedding using a difference expansion*. IEEE Transactions on Circuits and Systems for Video Technology, 13 : 890-896.

[4] Chen, W. J., Chang, C. C. and Le, T. H. N. 2010. *High payload stenography mechanism using hybrid engine detector*. Expert Systems with Applications, 37 : 3292-3301.

[5] Lee, C. F. and Chen, H. L. 2010. *A novel data hiding scheme based on modulus function*. Journals of Systems and Software, 83 : 832-843.

[6] Lee, C. F., Shen, J. J. and Lai, H. Y. 2018. *Data hiding using multi-pixel difference expansion*. Proceedings of the 3rd International Conference on Computer and Communication Systems, Nagoya, Japan, 27-30.

[7] Lee, C. F., Weng, C. Y. and Kao, C. Y. 2018. *Reversible data hiding using Lagrange interpolation for prediction-error expansion embedding*. Soft Computing.

[8] Lu, T. C., Wu, J. H. and Huang, C. C. 2015. *Dual-image-based reversible data hiding method using center folding strategy*. Signal Processing, 115 : 195-213.

[9] Malik, A., Sikka, G. and Verma, H. 2017. *An image interpolation based reversible data hiding scheme using pixel value adjusting feature*. Multimedia Tools and Applications, 77 : 13025-1304.

[10] Allebach, J. and Wong, P. W. 1996. *Edge directed interpolation*. Proceedings of 3rd IEEE International Conference on Image Processing, 707-710.

[11] Yang, C. N., Hsu, S. C. and Kim, C. S. 2017. *Improving stego image quality in image interpolation based data hiding*. Computer Standards and Interfaces, 50 : 209-215.

[12] Jung, K. H. and Yoo, K. Y. 2009. *Data hiding method using image interpolation*. Journal of Compute Standard and Interfaces, 31 : 465-470.

[13] Lee, C. F. and Huang, Y. L. 2012. *An efficient image interpolation increasing payload in reversible data hiding*. Expert Systems with Applications, 39 : 6712-6719.